# FINAL TECHNICAL REPORT

for Office of Naval Research Grant No.: NOOO14-95-1-0939

Don R. Brown, Ph.D. (PI)
Robert Mecklenburg (co-PI)
Department of Mechanical Engineering
3209 MEB, University of Utah
Salt Lake City, UT 84112
(801) 581-5369
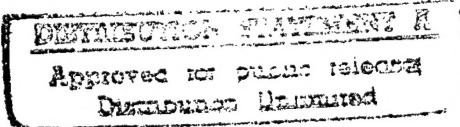
DEPARTMENT OF THE NAVY
OFFICE OF NAVAL RESEARCH
SEATTLE REGIONAL OFFICE
1107 NE 45TH STREET. SUITE 350
SEATTLE WA 98105-4631

IN REPLY REFER TO:

4330
ONR 247
11 Jul 97

From: Director, Office of Naval Research, Seattle Regional Office, 1107 NE 45th St., Suite 350, Seattle, WA 98105

To: Defense Technical Center, Attn: P. Mawby, 8725 John J. Kingman Rd., Suite 0944, Ft. Belvoir, VA 22060-6218

Subj: RETURNED GRANTEE/CONTRACTOR TECHNICAL REPORTS

1. This confirms our conversations of 27 Feb 97 and 11 Jul 97. Enclosed are a number of technical reports which were returned to our agency for lack of clear distribution availability statement. This confirms that all reports are unclassified and are "APPROVED FOR PUBLIC RELEASE" with no restrictions.

2. Please contact me if you require additional information. My e-mail is *silverr@onr.navy.mil* and my phone is (206) 625-3196.
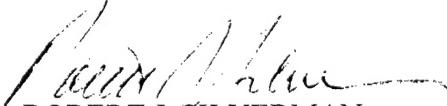
ROBERT J. SILVERMAN

INTRODUCTION

Managing military systems requires access to geographically distributed information. Difficulty in locating that information represents a serious threat to operational success.

Under this contract, the University of Utah has researched methods for the acquisition, display, and management of distributed information. Although the focus of the work is on product information, the research done here has broad applicability in supporting military operations management.

This research explored architectures and algorithms for distributed product information. The research was tested by incorporating these architectures and algorithms into an implemented system called PartNet.

PartNet is a scalable, client/server information system enabling mechanical and electronics component manufacturers and distributors to make their product catalogs available to customers over the Internet. This system allows catalog browsing capability, including multimedia product descriptions with computer aided design (CAD) and other analytical models. It interfaces to other systems with EDI or other protocols supporting purchase authorization, payment for products and data, order verification, shipment scheduling, and other functions necessary to support a supplier/customer interaction.

PartNet can satisfy a variety of significant customer needs. One example would aiding repair technicians in finding replacement parts for a defense system that has no part numbers available. PartNet would be used to perform supplier and part number identification. Another use would be to aid an engineer in finding information on a component for a system being designed or modified. The engineer could query PartNet for the technical specifications and dimensions of a component. Working from the desktop, the engineer would find candidate components, compare their specifications, and even retrieve CAD models if the supplier has provided them. Supply personnel would use PartNet to perform product and price comparisons and then purchase the component through PartNet's link to a purchasing system.

RESEARCH QUESTIONS

Several important questions were researched to determine whether a distributed system can ameliorate problems associated with product information retrieval.

Data integration: Is it possible to integrate the data of various vendors in such a way that meaningful product comparisons can be made?

Our research under this contract has shown that integration of data from multiple parts suppliers is practical, although in some cases somewhat difficult due to the varying conditions of supplier product data. Parts data from approximately a dozen suppliers are currently available for querying via PartNet.

Interactivity: Can the Internet support real-time interactive part browsing?

Tests conducted by engineers at the Sacramento Air Logistics Center have shown that real-time interactive parts browsing using PartNet is possible. Engineers at SM/ALC have used PartNet Client Software both the UNIX workstation version, and the newest MS-Windows version, to successfully locate parts used in SM/ALC work. Most recently, SM/ALC has been able to directly order the selected parts from the parts supplier via PartNet.

Scaleability: Can a federated database scale to the required number of vendors and customers?

In the past several months, the number of parts on-line through PartNet and the number of supplier databases participating in the PartNet project have grown significantly. PartNet has successfully accommodated these increases. There are currently a total of over 200,000 parts listed through PartNet.

PartNet's scaleability is achieved by the use of a Vendor Database Interface (VDI) software module at each participating supplier, communicating with PartNet Query Servers (QS). As the number of parts and suppliers grow, PartNet can expand by adding VDIs and QSs as needed. There are no hardware or software constraints on how large the network of participating parts suppliers can be with PartNet.

Data Accuracy: Can the system be easily updated so that information providers can maintain their own data?

During the term of this contract, Stock Drive Products/Sterling Instruments was enrolled as the first commercial parts supplier to offer parts data for use in making sales to customers through participation in the PartNet project. SDP/SI maintained its own product database on a computer located at its local (Long Island, NY) Internet provider. SDP/SI began with approximately 15,000 parts, then later increased the total number of parts accessible via PartNet to approximately 40,000.

From our initial experiences with SDP/SI, we learned that it is necessary to help the supplier prepare their data for use with PartNet. Once this initial difficulty is overcome, the supplier's dependence on technical support from PartNet diminishes significantly. This also proved to be the case with the similar, but much larger task of making approximately 125,000 parts available through Newark Electronics.

PartNet continues to research and develop software tools that will assist new suppliers in placing their electronic catalog data on PartNet.

Global Consistency: Can the data be made to be globally consistent as new data are added to the system?

If sufficient domain knowledge (e.g. the meanings of terms such as "RMS" or "hub spur gear") can be applied and relatively consistent data can be obtained from multiple vendors, then globally consistent parts taxonomy is possible.

Those prerequisites are sometimes difficult to obtain, since suppliers have historically developed their own in-house domain knowledge and were not anticipating that their in-house parts information database would ever by accessible by customers.

Developing common domain knowledge between suppliers is largely a function of the costs associated with the human resources that must be dedicated to developing a shareable set of information about a suppliers parts .

If domain knowledge is not supplied by with parts data, then PartNet programmers have to incorporate that knowledge into PartNet's algorithms.

Caching: To what extent can caching improve system performance? What kind of data should be cached and where should it be cached?

Data caching permits users to access frequently requested parts information without having to re-create complete searches of supplier databases. This significantly reduces response times.

Data should be cached at the Query Server (QS) to enable different users at a site quick access to the parts most frequently searched for from the users' site. Engineers working at different workstations in the same company or department can see the cached information without having to use the same workstation.

Caching at the QS reduces network traffic between the QS and the Vendor Database Interface (VDI). The time savings realized by caching varies with the types of parts most frequently searched and the network speed between the QS and the VDI.

To ensure accuracy of the information, the cache is automatiacally purged on a daily basis, if not more frequently, or any time the supplier's database has changed.

NATURE AND SCOPE OF RESEARCH

Method and Approach

PartNet is a project to provide direct, interactive on-line access to parts catalogs. This access relies on the Internet to provide an efficient communications medium for transferring parts information from vendors to customers. This approach has many advantages over both traditional paper catalogs and CD-ROM-based methods. Both paper and CD-ROM provide a more traditional "batch oriented" style of access to parts data. Normal manufacturing and production delays mean that customers cannot rely on this information to be up to daté or complete (due to space limitations). The discrete nature of paper and CD-ROM catalogs makes it impossible to search all catalogs simultaneously (without the number of disc drives being equal to the number of catalogs). It may also not be possible to acquire all catalogs, even from a single catalog distributor, due to shipping or publishing constraints (e.g., a new vendor has been added to a distributor's catalog suite, but you cannot get the catalog until the distributor's next product release).

The PartNet system overcomes these problems by providing high-speed access to all vendors simultaneously. All the information a vendor is willing to distribute is available, including CAD files, technical data sheets, photographic images, and animation. When a new vendor makes its database available to PartNet or when an existing vendor adds new products their information is immediately available to customers through the distributed PartNet software system.

The design of PartNet is driven by a small number of important issues:

First, it must be scalable to thousands of vendors and tens of thousands of customers. It should be possible to start with an initial installation of a single vendor and a few customers and grow from there. As the subscription rate increases the system should be dynamically configurable to handle the increased load.

Second, the system should be tolerant of network failure and processing delays. Since the system relies on databases maintained by vendors at the vendor's site the catalog information will be widely separated both geographically and "network-wise." If answering a query requires all vendor systems to be operational and timely then eventually no query could ever be answered.

Finally, the system must be "portable" in the most general sense of the word. Vendor databases all likely run on the full spectrum of computer hardware, use a wide variety of data base management system software (DBMS), and encompass many different data formats. All this diversity must be managed and translated into a unified format suitable for an on-line parts catalog.

One of the underlying assumptions of PartNet is that most vendors already have or will want to store their product information in on-line databases which are accessible to their customers. The PartNet system exports this product information in a controlled fashion to the customers who need it. Although this assumption may not reflect current business practice, we feel it is an obvious and economically sound choice.

Vendors must already maintain inventory and manufacturing databases; many also have design databases. These can be unified by a comprehensive product database which includes traditional catalog data, availability and delivery information, images, as well as non-traditional data such as animation or vendor tutorials. Information stored in on-line systems is easier to access, maintain and deliver to end users and will reduce the cost of delivery and dissemination.

Architecture

PartNet is designed as a heterogeneous, distributed system specialized for read-only access. Each vendor site presents a database of parts information available for access by customers. These databases are managed by varying (and possibly proprietary) database management systems. Vendor sites are distributed geographically as well. The Internet ties these systems together and PartNet software moderates communication through a common protocol.

Customers interact with the system through a PartNet client interface connected via the Internet to a centralized Query Server (QS). This Query Server receives queries about parts that are then routed to vendors who supply those parts. Each vendor site provides one or more Vendor Database Interface (VDI) processes that execute the query and return the answer to the QS. The QS in turn forwards the data to the original requesting customer. A single Query Server will handle a hundred or more customers simultaneously. As load increases Query Servers will be replicated.

This process architecture addresses several basic issues inherent to distributed databases:

First, the diversity of vendor database software is managed by a single coherent interface exported by the Vendor Database Interface (VDI). Each VDI is responsible for mapping from vendor specific formats into canonical PartNet format. This translation includes the DBMS query language, part attribute and value conversion, and image format conversion.

Second, the Query Server provides a centralized process for routing queries to vendors, managing global information to avoid inconsistent updates, and caching vendor data to reduce latency. The existence of the Query Server process also dramatically reduces the *NxM* connectivity problem (number of users multiplied by the number of suppliers) inherent with allowing customers to talk directly with vendors.

Without the Query Server, each vendor would need to establish and maintain connections with all users, and users, and each end user would need to independently establish connections with each Vendor Database Interface they wanted to submit queries to. Each end user would need to be aware of all VDIs, which is impossible, since VDIs may frequently appear and disappear from the network (due to maintenance, network problems, new vendors participating, etc.).

Third, the client software used by the customer is kept simple to allow execution on commonly-available MS-Windows operating systems or through World Wide Web browsers, such as Netscape.

Communication between processes is via a message-based command and response protocol. This allows a simple, portable implementation which is as efficient as Remote Procedure Call systems for average messages, but without the added implementation complexity.

The Vendor Database Interface

PartNet does not impose a particular DBMS or database management paradigm on participating vendors. This is important since vendors may have invested significant time and money into building their database. Furthermore, the vendor may even have a proprietary database management system tailored to their specific data. Any attempt to replace this database or impose an external standardized format will result in vendors who are unable or unwilling to participate in PartNet.

To avoid excluding vendors by requiring a standard database and query language, PartNet provides an interface process that responds to the PartNet communications protocol and implements database queries through native calls to the vendor database. This interface process manages network communication, taxonomy and names, concurrency, and caching.

We discuss each of these responsibilities in turn.

4

The first responsibility of the VDI is to manage network communication. Even if a vendor database directly accepted the PartNet command language, additional software would be required to identify the available Query Servers and manage network connections. When a VDI is initiated it identifies a Query Server through either a well known port and address or using the Internet name service. After establishing a connection to this Query Server it requests a complete list of active Query Servers with which it should register. By registering with a Query Server the vendor signals its readiness to receive and process queries. VDIs are able to accept connections from new Query Servers as they are added to the network and manage communication from Database Server Processes which are spawned to perform actual database queries.

Once a vendor has registered with a Query Server it transmits a taxonomy describing parts the vendor supplies and their relationship in the taxonomy of known mechanical parts. The Query Server receives this taxonomy and merges it with any existing taxonomy thus incrementally generating a global hierarchy of all parts known to the PartNet system. If the Query Server is presented with an as yet unknown part the VDI may be asked to send a detailed description of the part. This allows new parts to be added by vendors to the PartNet system.

The vendor must also present to the Query Server a list of synonyms used by customers and the Query Server to identify parts and their attributes. For instance, vendors may represent a part number as "PN,'' "part_number,'' "part_no,'' etc. Since the names of mechanical parts and their attributes is not standardized the PartNet system must be prepared to translate part names and attributes from a vendor specific value into a canonical form. This canonical form is used by the Query Server to uniquely identify parts and attributes and can be used in the customer user interface to simplify part selection and query formulation.

Names passed between the various components of PartNet always use the canonical name. There are two reasons for a VDI to transmit this table (since the Query Server has no use for it):

1. To provide this table to the customer for user interface reasons, and

2. To enable the Query Server to manage distributed updates to the shared synonym table.

If the user interface has this synonym table, customers can select parts using vendor specific nomenclature while still allowing the system to uniquely identify the part. Since the synonym table will be used to map from vendor's names to canonical names collisions must be managed (i.e., prevented or at least identified). This management will require global knowledge of all synonyms and a centralized change control capability (i.e., locking). This is done by the Query Server.

Since a VDI will be connected to several Query Servers which are in turn connected to many customers a vendor may be asked to answer several different queries in a small space of time. Ideally we would like to answer all queries immediately with response time related only to the delay imposed by the vendor's own DBMS. Unfortunately, there may be an arbitrary number of simultaneous queries limited only by the total number of customers. Also, many databases and operating systems are limited in the amount of concurrency a single program can achieve. For instance, a single program executing a database query might be required to wait until that query is processed by the DBMS and the answer returned before being allowed to initiate another query. This is overcome in the PartNet design by creating several database server processes which execute queries synchronously, but in parallel with each other.

The VDI process serializes all commands, but since each command can be handled very quickly (i.e., by forwarding the command to a Query Server or a database server) no command is forced to wait an undue amount of time for processing.

The final responsibility of the VDI is to aid in Query Server cache management. To improve throughput and reduce latency the Query Server caches answers to customer queries.

5

It is essential that a customer is never given obsolete information because the cache is inconsistent with the vendor's actual data. This is the problem of cache consistency and consequently, cache invalidation. To aid in maintaining a consistent cache the VDI must monitor the answers to queries it receives as long as the data is held in a Query Server cache. If this data ever changes the VDI must notify the appropriate Query Server that the original data is now invalid.

The Database Server

The Database Server is a slave process of the VDI and Query Server which performs actual database queries using the native DBMS interface. The purpose of this separate process is to overcome the singly threaded nature of many operating systems and DBMS interfaces. While a single Database Server process may perform one query at a time waiting for DBMS to process the query and return an answer, a collection of Database Server processes can handle multiple queries in parallel. These processes are managed by a single scheduler which maintains a queue of pending queries and a suite of available server processes. Queries are scheduled on idle processors and query answers are delivered using the standard PartNet protocol. It should be emphasized that this does not require a multiprocessor to execute. It is merely a mechanism to achieve process level parallelism in a singly threaded DBMS or operating system.

Although this does not achieve the ideal goal of the fastest query processing possible (which would require a CPU per query), it does provide a reasonable mechanism to maximize throughput and tune query processing. A simple algorithm would allocate a fixed number of Database Servers as determined by past query loads. A more sophisticated algorithm could dynamically adapt to query loads by spawning additional Database Servers as query arrival rate and system load dictate.

The Query Server

The Query Server is the "glue" which binds PartNet together. It provides a centralized service which can be accessed through either a well-known network address or by name from the Internet name server. Since it is centralized it forms a locus for routing information, global data management, and performance monitoring. We anticipate greater computational power at a Query Server host which can be used to reduce network traffic and latency through caching which may not be possible at the customer site (due to fewer computing resources).

The major responsibilities of the Query Server are:

1. Manage a set of customers and vendors
2. Route messages from customers and vendors
3. Control access to global data (taxonomy, parts, synonyms)
4. Cache data
5. Log transactions

As the central router for messages the Query Server must ensure that each customer is serviced fairly and that no customer process is "orphaned" or "mislead." In particular, answers to customer queries are delivered to the customer incrementally as each vendor supplies their portion of the answer. It is important that the customer not mistake a partial answer for a complete one. The PartNet client software informs the user that the answers being received represent responses from "X" of "Y" vendors identified as having parts matching the user's search criteria. When "X" equals "Y" the user knows that all responses have been received.

For each query submitted by a customer the Query Server determines which vendor is capable of supplying an answer and forwards the query to that set of capable vendors. This dramatically reduces network traffic when compared with forwarding every query to every vendor. To properly determine the capable vendors the Query Server must know all parts supplied by each vendor and it must update this information as it changes.

6

Other information the Query Server manages is global data such as the taxonomy, parts list, and synonym table. These items are global in that they unify all information supplied by all vendors with each vendor supplying their portion. A problem arises when two vendors wish to update this global data simultaneously. To properly handle this case some sort of concurrency control is required. PartNet uses an optimistic locking algorithm which allows any vendor to modify global information and request an update at the Server. When the Server receives the update request it determines if the update is valid. If not, the vendor's request is rejected and the vendor must retry the update.

To improve throughput and reduce latency the Query Server caches the answers to previous queries. When a query is received from a customer the cache is first scanned for other queries about the same part requested in the current query. If any are found the cached query is analyzed to determine if the previous query describes a superset of the current query. If this is true the current query can be answered directly from the cache without the overhead of forwarding the query to the vendors.

If a cache is employed the problem of cache consistency and invalidation must be solved. In short, a problem occurs when a vendor updates part information while the Query Server has cached information about that part. In this case the customer may be given out of date information about a part. This problem is solved by requiring the VDI process to inform the query server whenever parts information is changed. To reduce the burden on the VDI and reduce network traffic the Query Server associates a lifetime with each answer. When the lifetime has expired the answer is removed from the cache. This lifetime should be long enough to allow reasonable performance gains while short enough to minimize the load on the network and VDI.

Finally, the Query Server is responsible for monitoring the performance of the PartNet system as a whole. This includes ensuring that vendors and customers are not "orphaned", recording timing statistics on network latency and bandwidth, recording quantity of information delivered by each vendor to each customer (e.g., for billing purposes), and recording general usage patterns. Due to faults in networks and software it may be possible for customers or vendors to become unreachable. This should be noted and should not cause other software (e.g., Query Server or customer interface) to fail. Also, by recording network performance statistics the Query Server can improve the user interface by anticipating delays.

The User Interface

After initially developing client software for UNIX workstations, we became aware that in order to achieve the greatest possible accessibility to PartNet, client software for Windows-based PCs needed to be developed. This was accomplished after the expiration of the contract, but prior to the writing of this final report.

An example of version 1.0 of the client software conducting a search through PartNet is shown below:

The following illustrations are screen captures taken from the new PartNet client software for Windows showing the typical process for identifying and ordering a part. In this example, the client will be searching for a particular Hydraulic Circuit Breaker.

Illustration #1 (following page) shows the initial PartNet client software window, containing a list of available parts categories.

The "Circuit Protection Devices" category has been opened to reveal the sub-categories contained in this category. The "Circuit Breakers" sub-category has similarly been expanded to show the types of Circuit Breakers available for search. The "Hydraulic" sub-category has been selected.

Once the desired part sub-category has been identified and selected, the user selects the "Create Query" command from the software's Query Menu.
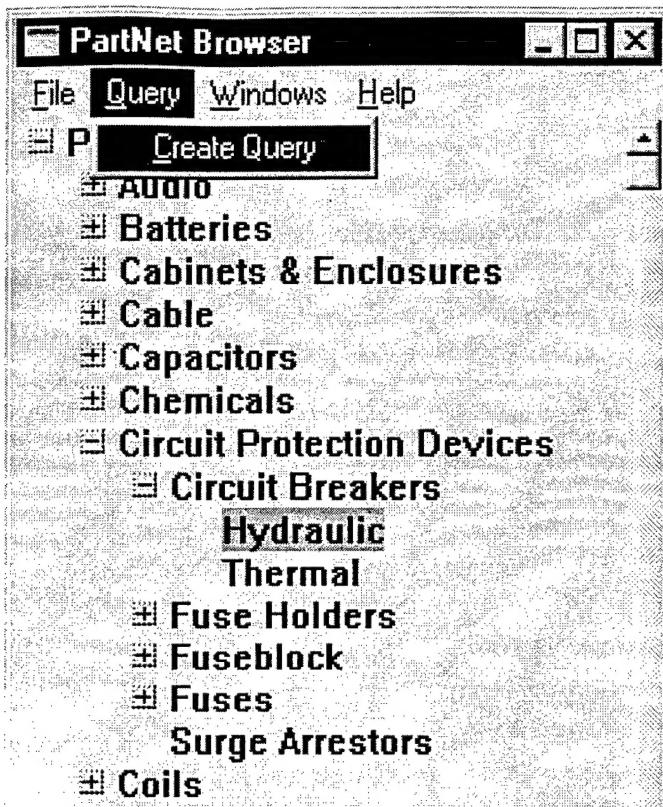
#1: The PartNet List of Parts Categories and Sub-Categories

#2: PartNet Query Window (unused attribute field rows have been deleted to save space)

Illustration #2 shows the PartNet Query Window with the desired part's Vender Part Number entered into the widow's Value field. Since the part's Vendor Part Number is known, it is not necessary to search for the part using a Boolean search of technical attributes. PartNet does support such searches where technical attributes are included in the parts data. As many as seven attributes may be incorporated into a PartNet query.

Part Detail

Hydraulic

| Attribute	Units	Value
Distributor Part Number		89F5593
Distributor		Newark
Vendor Part Number		BA1-B034630121C
Unit of Measure		EA0001
Vendor Name		CARLINGSWITCH
Minimum Order		1
Catalog Page Number		0
Quantity on Hand		21
Status		active
Catalog Status		stocked, in catalog
Multiple Sell Quantity		1
Price 1	(dollars)	17.16
Price 1 Quantity		1 - 9
Price 2	(dollars)	13.09
Price 2 Quantity		10 - 24
Price 3	(dollars)	10.43
Price 3 Quantity		25 - 49

[Close]

#3: Detail Window Returned by PartNet

Illustration #3 shows the "Detail Window" returned by PartNet after a successful search for the selected part. The Detail Window contains a list of the attribute information available for the selected part.

Additional files relating to a part, such as GIF images, CAD models, or technical description data sheets, are often available for downloading to the user via PartNet. These additional files are accessible for many parts listed through PartNet, but have not yet been added to the Newark Electronics database used in this project.

After PartNet has responded to the user's query and returned a listing for the desired part, the user has the option of placing an order for the selected part through the FAST electronic brokering system (Illustration #4)
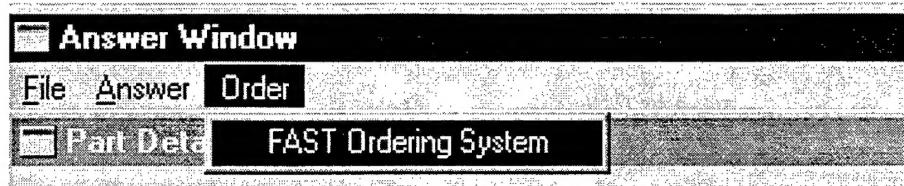
#4: Launching the FAST On-line Ordering System

When launched from within PartNet, the FAST system asks the user for a User ID and Password (Illustration #5). In addition to authenticating the user, this also identifies the user's payment method and shipping address.
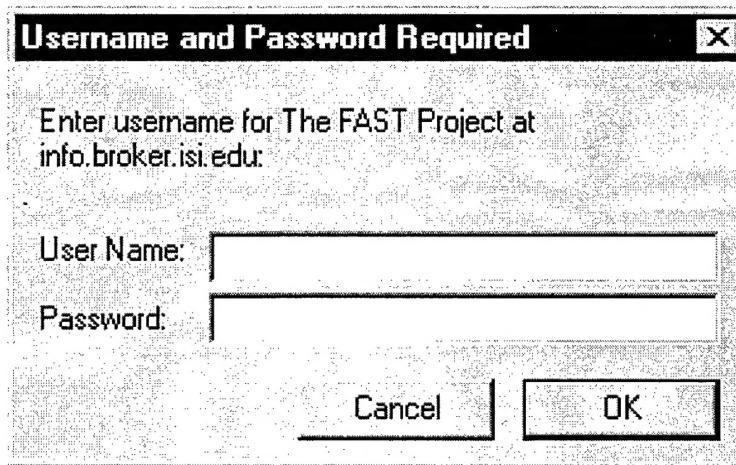
#5: FAST Log-In Area

After logging-in with FAST, the user is given an order form on which information required to process the order is entered (Illustration #6).

PartNet Catalog Data

Mfr: CARLINGSWITCH
Mfr-PN: BA1-B034630121C
Vendor: Newark
Vendor-PN: 89F5593
Unit-Price: $17.16 Unit-of-Measure: EA 0001

Mandatory Order Entry Fields

Quantity: [   ] (in the units shown above)

Order-No: [   ] (must be unique, max width 14)

Additional Identification and Tracking

NSN: [   ] - [   ] (for your reference)

Item-Label: [   ] (appears on packing slip)

Request-Label: [   ] (additional request tracking)

Funding-Code: [   ]

Deadlines and Shipping Information

(Dates may be entered DD-MON-YY, MM/DD/YY, or YY.MM.DD)

Ship-After: [   ]   Ship-Before: [   ]

Ship-To: [MARSHAD ▼] Ship-Via: [GROUND ▼]

Partial-Shipments: ○ Allowed ● Disallowed

Send the Request to FAST

Press [Send to FAST   ] to send this request to FAST.

#6: FAST Order Form

The other interface discussed is that of the World Wide Web. Here we have extended the Query Server command set to allow the Web to contact the Query Server directly to access the taxonomy and identify vendors. The Web software can then access the vendor catalog directly through a hypertext link using Web browser such as Netscape. The PartNet system may be accessed on the Web at http://www.part.net.

Methodology & Work Plan

The University of Utah designed and implemented the PartNet system consisting of the following subsystems:

Client software installed at the Sacramento Air Logistics Command.

A Query Server running on a file server at the University of Utah.

A Vendor Database Interface running on a dealer's site.

The University of Utah briefed additional customer sites and installed clients at those locations, such as the Defense electronics Supply Center and one other site. (Sacramento Air Logistics Center already has client installed.)

The University of Utah visited DoD customer sites including the Sacramento Air Logistics Center and the Defense Electronics Supply Center to brief users about PartNet. Contacts were also made to recruit new participating parts suppliers.

PartNet makes extensive use of object-oriented programming techniques to make maximum re-use of code. Less than 10% of PartNet code is specific to a given executable, such as the Query Server or Vendor Database Interface. The remaining code resides in common libraries for use/re-use by the executables.

Object-oriented techniques were extended to communications between the programs. Communications are done by the serialization of objects which are passed via TCP/IP connections and the original objects are re-created in the receiving executable.

Proprietary Claims

The University of Utah claims proprietary rights to any source and object code produced as a byproduct of this proposal. The University of Utah will grant a non-exclusive royalty-free license of this source and object code to the United States Department of Defense and its agencies for their own internal use if this proposal is accepted and funded.